

## IMPORTANT PROBLEM TYPES

The most important problem types are:

- (i). Sorting.
- (ii). Searching
- (iii). String processing
- (iv). Graph problems
- (v). Combinatorial problems
- (vi). Geometric problems
- (vii). Numerical problems.

### (i) Sorting

- The *sorting problem* is to rearrange the items of a given list in non- decreasing (ascending) order.
- Sorting can be done on numbers, characters, strings or records.
- To sort student records in alphabetical order of names or by student number or by student grade-point average. Such a specially chosen piece of information is called a *key*.
- An algorithm is said to be **in-place** if it does not require extra memory, E.g., Quicksort.
- A sorting algorithm is called **stable** if it preserves the relative order of any two equal elements in its input.

### (ii) Searching

- The *searching problem* deals with finding a given value, called a *search key*, in a given set.
- E.g., Ordinary Linear search and fast binary search.

### (iii) String processing

- A *string* is a sequence of characters from an alphabet.
- Strings comprise letters, numbers, and special characters; bit strings, which comprise

zeros and ones; and gene sequences, which can be modeled by strings of characters from the four-character alphabet {A, C, G, T}. It is very useful in bioinformatics.

- Searching for a given word in a text is called string matching

#### (iv) Graph problems

- A **graph** is a collection of points called vertices, some of which are connected by line segments called edges.
- Some of the graph problems are graph traversal, shortest path algorithm, topological sort, traveling salesman problem and the graph-coloring problem and soon.

#### (v) Combinational problems

- These are problems that ask, explicitly or implicitly, to find a combinational object such as a permutation, a combination, or a subset that satisfies certain constraints.
- A desired combinatorial object may also be required to have some additional property such as a maximum value or a minimum cost.
- In practical, the combinatorial problems are the most difficult problems in computing.
- The traveling salesman problem and the graph coloring problem are examples of **combinatorial problems**.

#### (vi) Geometric problems

- **Geometric algorithms** deal with geometric objects such as points, lines, and polygons.
- Geometric algorithms are used in computer graphics, robotics, and tomography.
- The **closest-pair problem** and the **convex-hull problem** are coming under this category.

#### (vii) Numerical problems

- **Numerical problems** are problems that involve mathematical equations, systems of equations, computing definite integrals, evaluating functions, and soon.
- The majority of such mathematical problems can be solved only approximately.